



## IV Valeurs booléennes :0,1. Opérateurs booléens.

En informatique, un booléen est une variable à deux états (souvent appelés vrai ou faux).

On rencontre des valeurs booléennes quotidiennement : Par exemple lors d'une tentative de communication téléphonique. Pour qu'elle ait lieu, il faut que Emetteur et Récepteur soient à « vrai ».

Cet état est souvent symbolisé informatiquement par 1 (Vrai ) ou 0 ( Faux)

Les valeurs booléennes sont souvent utilisées dans le cas de test conditionnel en programmation.

### 1) Opérateurs booléens

Les variables booléennes peuvent être reliées entre elles à l'aide d'opérateurs : **AND**(et), **NOT**(non) , **OR**(ou), **XOR** (ou exclusif, c'est-à-dire , l'un ou l'autre mais pas les deux simultanément)

*Différenciation des deux « ou » : Le premier ou (OR) correspond à celui utilisé dans la phrase « Je viendrai s'il ya un bus ou un taxi ». Le second,(XOR), correspond à : « J'irai à la plage ou à la montagne ».*

*Avec OR, la fonction booléenne  $OR(a,b)$  vaut 1 si **au moins** une des deux valeurs de a ou de b vaut 1.*

Avec XOR, la fonction booléenne  $XOR(a,b)$  vaut 1 si **uniquement** une des deux valeurs de a ou de b vaut 1.

A l'aide de ces opérateurs, on peut dresser des tables d'expression booléenne.

a	b	NON a	NON b	a AND b	a OR b	a XOR b
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	1	0

## 2) Opérateurs booléens et addition binaire

On a vu comment additionner deux nombres en binaire. Il existe un lien avec les opérateurs booléens.

A	B	RETENUE R	SOMME S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Quel opérateur permet d'obtenir R ? S ?

	<p>Les opérateurs <i>AND</i> et <i>OR</i> sont séquentiels : si l'évaluation de la première opérande suffit à donner le résultat, la deuxième n'est pas évaluée. Ainsi, si <math>A=0</math>, <math>A \text{ AND } B=0</math>, sans que B n'ait été évalué.</p>
-------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 3) Expressions booléennes

Voici un exemple de fonction booléenne : La **fonction multiplexeur**, notée mux.

$$\text{mux}(x,y,z)=(\text{not}(x) \text{ and } y) \text{ or } (x \text{ and } z)$$

x	y	z	not(x)	not(x) and y	x and z	mux(x,y,z)
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Remarque : On peut exprimer toutes les fonctions booléennes à l'aide de *and*, *or* et *not* [4\) Exercices](#)

1) Montrer que  $(x \text{ and } y) = \text{not}(\text{not}(x) \text{ or } \text{not}(y))$

2) Montrer que  $(x \text{ or } y) = \text{not}(\text{not}(x) \text{ and } \text{not}(y))$

3) Trouver l'expression de la fonction ssi (x,y) à l'aide des opérateurs booléens.

x	y	ssi(x,y)
0	0	1
0	1	0
1	0	0
1	1	1

4) Ecrire des fonctions en langage python correspondant aux opérateurs **and**, **or**.

*and* et *or*, ainsi que *not*, sont des instructions en langages python qui fonctionnent comme les opérateurs booléens.

```
def booleen(a,b):  
    return( a and b)  
  
>>> booleen(0,1)  
0  
>>> booleen(0,0)  
0  
>>> booleen(1,0)  
0  
>>> booleen(1,1)  
1  
.
```